
Cognitive Discriminative Mappings for Rapid Learning

Wen-Chieh Fang

Yi-ting Chiang

Abstract

Humans can learn concepts or recognize items from just a handful of examples, while machines require many more samples to perform the same task. In this paper, we build a computational model to investigate the possibility of this kind of rapid learning. The proposed method aims to improve the learning task of input from sensory memory by leveraging the information retrieved from long-term memory. We present a simple and intuitive technique called cognitive discriminative mappings (CDM) to explore the cognitive problem. First, CDM separates and clusters the data instances retrieved from long-term memory into distinct classes with a discrimination method in working memory when a sensory input triggers the algorithm. CDM then maps each sensory data instance to be as close as possible to the median point of the data group with the same class. The experimental results demonstrate that the CDM approach is effective for learning the discriminative features of supervised classifications with few training sensory input instances.

1 Introduction

Scientists have interest in understanding the relations between all the levels that describe what a brain does. Adolphs listed the top 23 unsolved problems in neuroscience (including three "meta" issues) [1]. Two important questions, "How does sensory transduction work?" and "How does learning and memory work?", are closely connected [1]. Without sensory transduction that converts a sensory stimulus from one form to another, the brain cannot integrate and process the sensory input information with information stored in the memory.

People can learn a new concept or recognize an item from just a handful of examples, while state-of-the-art machine learning algorithms typically require tens or hundreds of examples to perform with similar accuracy [6]. Scientists have long suspected that this type of "one-shot learning," or rapid learning, involves a different mechanism in the brain than gradual learning [12]. We believe the underlying mechanism contains a complex information processing procedure.

Humans have five main senses: sight, hearing, taste, smell, and touch [9]. The human brain can combine data from sensory memory (SM) and prior experiences retrieved from long-term memory (LTM) into a single phenomenal experience. *Feature integration theory* tackles the question of how humans perceive individual features as part of the same object by proposing a two-stage process: preattentive processing and focused attention processing [10] [11]. The basic idea is that objects are analyzed into features and that attention is necessary to combine these features to create perceptions of an object. We apply this theory to our model to describe how sensory input data are processed in SM. In addition, people can retrieve prior experiences encoded in LTM. We believe that this procedure is one of the reasons why humans learn new concepts quickly. Therefore, we simulate this procedure in our model.

In this paper, we model rapid learning as a prediction task with feature transformation and augmentation procedures in the working memory. The rapid learning problem aims at achieving high prediction

performance when training a learning system with limited data drawn from SM by leveraging a relatively large amount of data retrieved from LTM.

In Section 2, we formally define the problem and introduce the framework of our approach. Then we present our approach, including the learning of discriminative features and the augmentation techniques in Section 3. We proceed by evaluating our approach on benchmark data sets and discuss the results in Section 4. Finally in Section 5, we conclude the paper and offer suggestions for future work.

2 Problem Definition

Assuming that we have a labeled data set of N_L points: $\{(\mathbf{x}_i, l_i)\}_{i=1}^{N_L}$. The class l_i of each data instance $\mathbf{x}_i \in \mathbb{R}^m$ is in class set C_L . We denote this data set as LTM data set D_L . Let there be another data set of interest. We denote it as SM data set D_S , which has few labeled data instances $\{(\mathbf{y}_i, l_i)\}_{i=1}^{N_S}$. Each instance $\mathbf{y}_i \in \mathbb{R}^n$. The class l_i is in class set C_S . The class set C_S is assumed to be equal to C_L . The LTM and SM data sets are in different feature spaces. In most cases, the number of dimensions of the two input data sets are also different.

Our goal is to build a classifier by making use of the small data $\{(\mathbf{y}_i, l_i)\}_{i=1}^{N_S}$ from SM and the relatively large data $\{(\mathbf{x}_i, l_i)\}_{i=1}^{N_L}$ from LTM. The classifier can be applied to new incoming data in SM and performs equally or even better than the classifier trained on data from SM. Note that there is no data instance without knowing its class from SM during the training phase.

According to the definition in [7], given two domains, LTM domain $\mathcal{D}_L = \{\mathcal{X}, P(X)\}$ and SM domain $\mathcal{D}_S = \{\mathcal{Y}, P(Y)\}$, where \mathcal{X}, \mathcal{Y} are feature spaces and $P(X), P(Y)$ are marginal probability distributions, $X = \{\mathbf{x}_i\}_{i=1}^{N_L} \in \mathcal{X}$ and $Y = \{\mathbf{y}_i\}_{i=1}^{N_S} \in \mathcal{Y}$. If $\mathcal{D}_L \neq \mathcal{D}_S$, this implies that either $\mathcal{X} \neq \mathcal{Y}$ or $P(X) \neq P(Y)$. In this paper, we focus on the situation of $\mathcal{X} \neq \mathcal{Y}$ and $C_L = C_S$.

We give two definitions in our problem:

1. A *discriminative cluster* is a group of objects that share the same class label in a metric space.
2. The *radius of a cluster* is the maximum distance between all the points and the median point.

We also give some assumptions:

1. Data from SM and LTM exist for each class. Therefore our approach is a kind of supervised method.
2. The data from SM and LTM may share *no* co-occurrence features. We do not rely on co-occurrence features to train the model; therefore, our approach can extend well to heterogeneous domains.
3. No instance is shared across domains.
4. The relationship between the LTM domain and the SM domain is not given. We only know that the two domains have the same classes in common.
5. The new incoming data from SM for evaluation are unseen during the training phase.

2.1 Lemma

Lemma 1. A set of clusters C is said to be pairwise disjoint if and only if for every $C_L, C_S \in C$, let r_{C_L} and r_{C_S} be the radii of C_L and C_S respectively; then

$$d(C_L, C_S) > r_{C_L} + r_{C_S}, \quad (1)$$

where $d(C_L, C_S)$ is the distance between the median points of the two clusters C_L and C_S .

2.2 Hypothesis

We present the following main theoretical hypothesis. We believe the proposed hypothesis provides a promising theoretical base for us to develop the algorithm.

Hypothesis 1. Given that there exists an SM domain \mathcal{D}_S , a sample D_S is from \mathcal{D}_S . If we can find a sample set D_L in the LTM domain \mathcal{D}_L , and two mapping functions f and g are such that the following conditions are satisfied:

1. Each data instance $\mathbf{x}_i \in D_L$ with class l and each data instance $\mathbf{y}_j \in D_S$ with same class l are mapped into a common discriminative cluster corresponding to the same class l in a new space.
2. These discriminative clusters are pairwise disjoint.

Then, there exists at least a hypothesis $h \in H$ (H is a family of hypotheses in \mathcal{D}_S) that the empirical error rate

$$\hat{\epsilon}_{\mathcal{U} \cup \mathcal{V}}(h) \leq \hat{\epsilon}_{\mathcal{V}}(h) \quad (2)$$

where \mathcal{U} is the projected LTM sample set in the new space, and \mathcal{V} is the projected SM sample set in the new space. In other words, the samples in \mathcal{U} help to reduce the empirical error rate.

3 Proposed Solution

3.1 Main Idea

Because the data from LTM and SM are in different feature spaces, it is desirable to find a common invariant feature space in which all data can be directly compared. Inspired by the *min-max principle* [5] and *class-based constraints* [8], we apply two transformations to map the LTM and SM data in order, to satisfy class constraints between the transformed points.

In order to learn two transformation functions f and g for the SM data classification, we first define two variables ψ_S and ψ_D as follows:

$$\begin{aligned} \psi_S(f, g) &= \sum_{i \in \text{LTM}, j \in \text{SM}, l_i = l_j} d_{\Omega}(f(\mathbf{x}_i), g(\mathbf{y}_j)) + \sum_{i, j \in \text{LTM}, l_i = l_j} d_{\Omega}(f(\mathbf{x}_i), f(\mathbf{x}_j)) + \sum_{i, j \in \text{SM}, l_i = l_j} d_{\Omega}(g(\mathbf{y}_i), g(\mathbf{y}_j)) \\ \psi_D(f, g) &= \sum_{i \in \text{LTM}, j \in \text{SM}, l_i \neq l_j} d_{\Omega}(f(\mathbf{x}_i), g(\mathbf{y}_j)) + \sum_{i, j \in \text{LTM}, l_i \neq l_j} d_{\Omega}(f(\mathbf{x}_i), f(\mathbf{x}_j)) + \sum_{i, j \in \text{SM}, l_i \neq l_j} d_{\Omega}(g(\mathbf{y}_i), g(\mathbf{y}_j)) \end{aligned} \quad (3)$$

Here, $d_{\Omega}(\cdot, \cdot)$ is the distance function defined in the common space Ω . ψ_S sums the distance between the transformed points from LTM and SM with the *same* class, while ψ_D indicates the sum of the distance between the projected instances from LTM and SM with *different* classes.

Then, we posit that there exist two bounds u and l such that two inequations are necessary for the generation of pairwise disjoint discriminative clusters:

$$\begin{aligned} \psi_S(f, g) &\leq u \\ \psi_D(f, g) &\geq l \end{aligned} \quad (4)$$

where u and l are the upper and lower bounds chosen so that ψ_S should be small, and ψ_D should be large. Given the LTM data, SM data, and the class constraints in advance, if we can learn two transformation functions f and g from the data to satisfy inequations 4, we are able to use LTM data to help improve prediction performance.

3.2 Our Solution

In this paper, we present a linear version. We represent f and g as two *linear* transformation functions, namely \mathbf{P} and \mathbf{Q} . That is, $f(\mathbf{x}_i) = \mathbf{P}\mathbf{x}_i$ and $g(\mathbf{y}_j) = \mathbf{Q}\mathbf{y}_j$.

We present a computational mappings model to make use of both data from LTM and SM. At first, sensory data instances are being taken in by sensory receptors and kept in SM. When the sensory data instances are stored in SM long enough, they are transferred to working memory. In the new working memory space, these transformed data are populated into clusters according to their classes. Fig. 1 illustrates the main idea of the proposed model.

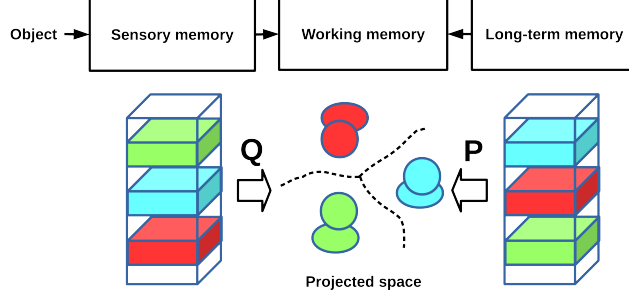


Figure 1: Illustration of Cognitive Discriminative Mappings, in which different colors represent different classes. The broken black lines represent the class boundaries. Circles represent the grouped instances from LTM while ovals represent the projected instances from SM. If the class boundaries can discriminatively separate instances into different classes, low classification errors will be expected.

We can estimate the two linear transformations \mathbf{P} and \mathbf{Q} by first deciding one transformation. Most of the time, there are many more related samples from LTM than there are sensory cases in SM. Consequently, we use mapping function $\mathbf{P} : \mathbf{X} \rightarrow \mathbf{U}$ to project the LTM data $\mathbf{X} \in \mathbb{R}^{N_{\mathcal{L}} \times m}$ into $\mathbf{U} \in \mathbb{R}^{N_{\mathcal{L}} \times d}$ in a latent space Ω in working memory. Therefore $\mathbf{P} \in \mathbb{R}^{d \times m}$. The latent space Ω is assumed to be a d -dimensional Riemannian manifold.

We present three different approaches in the retrieval phase. The first approach is where the c geometric medians of clusters corresponding to the classes are fixed and predefined in the latent space. The data in LTM are mapped into these locations by a transformation matrix \mathbf{P} and form discriminative clusters simultaneously.

In the second approach, we apply Linear discriminant analysis (LDA) [4] to characterize or separate the classes of instances. We use the transformation matrix \mathbf{P} provided by LDA to project original instances onto a latent space, and a new feature set is generated.

In the third approach, a linear transformation \mathbf{P} is derived by minimizing a cost function similar to Graph embedding method (GE) [13] as follows:

$$\mathbf{P} = \min_{\mathbf{P}} \frac{1}{2} \sum_{i,j} \|\mathbf{P}\mathbf{x}_i - \mathbf{P}\mathbf{x}_j\|^2 \mathbf{W}_{ij}, \quad (5)$$

where \mathbf{W} is a sparse symmetric $N_{\mathcal{L}} \times N_{\mathcal{L}}$ matrix that represents class similarity relationship and where $\mathbf{W}_{ij} = 1$ if $l_i = l_j$ and $\mathbf{W}_{ij} = -1$ if $l_i \neq l_j$.

Given the transformation \mathbf{P} of the data from LTM, we can find a mapping $\mathbf{Q} : \mathbf{Y} \rightarrow \mathbf{V}$ such that each data instance $\mathbf{y}_i \in \mathbb{R}^n$ with class $l_i \in C$ is mapped to a point $\mathbf{v}_i \in \mathbb{R}^d$ in the neighborhood of the cluster corresponding to the class l_i . In this paper, we provide an approximation solution to find such a mapping. The idea is that each data instance \mathbf{y}_i with class label $l \in C$ is mapped to a point \mathbf{v}_i such that \mathbf{v}_i is as close as possible to the median point γ_l . We can find such a linear mapping matrix $\mathbf{Q} \in \mathbb{R}^{d \times n}$ by setting:

$$\mathbf{Q} = \arg \min_{\mathbf{Q}} \sum_l \sum_{i:l_i=l} \|\mathbf{Q}\mathbf{y}_i - \gamma_l\|^2, \quad (6)$$

where $\sum_{i:l_i=l}$ denotes the summation over i such that label $l_i = l$.

To avoid overfitting, we add a regularization term $\zeta(\mathbf{Q})$. For example, we can set $\zeta(\mathbf{Q}) = \eta \|\mathbf{Q}\|_F^2$, in which η is the weight and $\|\cdot\|_F$ is the Frobenius norm. The \mathbf{P} and \mathbf{Q} mapping functions are derived for the purpose of obtaining a low-dimensional representation of the data that separates the populations as much as possible. However, the classes may not be sufficiently separated. Consequently, we apply a discrimination method again to determine a mapping function \mathbf{H} to render the classes as separated as possible.

3.3 Feature Augmentation

After applying the final discrimination method to transfer the data, we derive new features. Daumé III provides a *feature augmentation* method to integrate more information [3]. For each instance, we utilize the feature augmentation method to augment the new features with the original features to represent the instance [3].

We define two feature mapping functions $\phi_L(\mathbf{x}) = [(\mathbf{H}\mathbf{P}\mathbf{x})^\top, \mathbf{0}_n^\top]^\top$ and $\phi_S(\mathbf{y}) = [(\mathbf{H}\mathbf{Q}\mathbf{y})^\top, \mathbf{y}^\top]^\top$ for the data from LTM and SM, respectively. Here $\mathbf{0}_n$ denotes zero column vectors of dimensions n . Including zeros in the feature representations ensures that the dimensions of the instances from LTM and SM become the same. Moreover, the entire data set from LTM has no information on the features from SM. Therefore, it is reasonable to set zero values ($\mathbf{0}_n^\top$) in the later n feature dimensions.

3.4 Proposed Algorithmic Procedure

As a summary, we give the algorithm of the proposed method:

1. Use a projection approach to learn a mapping \mathbf{P} to project data retrieved from LTM to a new d dimensional space in the working memory. All the projected data in the new space are grouped into several clusters corresponding to the classes.
2. Compute the generalized geometric median of all clusters.
3. Compute the mapping function \mathbf{Q} for data in the SM via these generalized geometric medians.
4. Apply \mathbf{Q} to map the SM data for training and new incoming SM data to the new d dimensional space.
5. Apply a discrimination method again to construct a transformation matrix \mathbf{H} to separate *all* the instances into distinct classes.
6. Augment the learned features with the original SM features to represent the instance.
7. With the projected data with known classes from LTM and SM as training data, conventional supervised machine learning methods can be used to learn the model and predict the projected new incoming data from SM.

4 Experimental Results

In this section, we conduct several experiments on two benchmark data sets. We assume that there is only one data set from LTM and one data set from SM. Because the upper bound of the number of retained dimensions in LDA is $c - 1$ (c is the number of classes), we set the dimension of the new space in the first projection for learning \mathbf{P} and final projection to $c - 1$.

For selection of the weight of the regularization term η , cross-validation is not applicable due to the small number of training instances from the SM domain. Therefore, we tune the parameters on a predefined range and report the optimal parameter value. In this paper, we report the results when $\eta = 1$.

We apply a k nearest neighbor classifier (kNN, $k = 5$) and a Support Vector Machine with a Radial Basis Function kernel (RBF SVM) to train the final classifiers on the data sets. The training instances are derived from both LTM and SM. For the baseline approach, we train the classifier on the original labelled instances from SM and use it to predict the test instances. We randomly sample the training instances ten times and report the average accuracy over the ten rounds of experiments. In this paper, we conduct experiments in different settings in the two data sets and report the results.

4.1 Data Sets

4.1.1 Object Recognition Data Set

The first data set¹ contains 4,652 images from 31 categories² originating from the following three domains: Amazon (images downloaded from an online retail website), `dslr` (high-resolution images taken from a digital DLR camera) and `webcam` (low-resolution images taken from a web camera) [8]. SURF features are extracted for all the images.

The Amazon and `webcam` images are used as data retrieved from LTM. We randomly select twenty and eight training images per category for the Amazon and `webcam` data sets, respectively. We then randomly select k training images per category for the `dslr` data set as SM data, where $k = 3, 4, 5$, and 6. The remaining images are used for testing. Table 1 shows a summary of the data set.

Table 1: Summarization of the object recognition data set.

	Data set	# total instances	# dim	# training data instances per class
LTM	Amazon	2813	800	20
	webcam	795	800	8
SM	<code>dslr</code>	498	600	3/4/5/6

4.1.2 Text Categorization Data Set

The second data set is a subset of the Reuters RCV1/RCV2 collections [2]. It contains newswire articles written in English, French, German, Italian, and Spanish. There are six classes for these articles: C15, CCAT, E21, ECAT, GCAT, and M11.

We take Spanish articles as SM data and articles written in the other four languages as individual data sets retrieved from LTM. For each class, we randomly sample one hundred training instances from the data sets in LTM and k training instances from SM data, where $k = 5, 7, 10, 15$, and 20. We also perform a Principal Components Analysis (PCA) with 60% energy preserved on the TF-IDF features. The remaining instances from SM data are used as the test instances. Table 2 shows a summary of the data set, and Table 3 shows the distribution of classes in the data set.

Table 2: Summarization of the text categorization data set

	Data set	# dim after PCA	# total instances	# training data instances per class
LTM	English	1,131	18,758	100
	French	1,230	26,648	
	German	1,417	29,953	
	Italian	1,041	24,039	
SM	Spanish	807	12,342	5/7/10/15/20

Table 3: Distribution of classes in the text categorization data set

Class	Size(all languages)	(%)
C15	18,816	16.84
CCAT	21,426	19.17
E21	13,701	12.26
ECAT	19,198	17.18
GCAT	19,178	17.16
M11	19,421	17.39

4.2 Comparative Studies

4.2.1 Object Recognition Data Set

Table 4 shows the performance of the baseline approach and the CDM approach in different settings on the object recognition dataset. The results show that the CDM approach using a kNN classifier both with and without augmentation have outstanding performance in the object recognition dataset. In the case of the RBF SVM classifier, although the learned features are not good enough for training a prediction model, augmenting the learned features with the original features from SM improves the performance.

¹Visit <http://vision.cs.uml.edu/adaptation.html> for more details.

²These 31 categories are: backpack, bike, bike helmet, bookcase, bottle, calculator, computer, desk chair, desk lamp, file cabinet, headphones, keyboard, laptop, letter tray, mobile phone, monitor, mouse, mug, notebook, pen, phone, printer, projector, puncher, ring binder, ruler, scissors, speaker, stapler, tape, and trash can

Table 4: Means and standard deviations of classification accuracies (%) of the baseline and the CDM approaches on the object recognition dataset by using 3 labeled training samples per class from the SM domain dslr. The CDM approach shows the results in different settings.

LTM	Classifier	Baseline	Without augmentation	With augmentation
Amazon	kNN	32.6 ± 2.3	45.4 ± 2.2	44.0 ± 2.3
	SVM	50.6 ± 3.0	39.4 ± 3.4	54.2 ± 3.0
webcam	kNN	32.6 ± 2.3	42.9 ± 2.7	42.6 ± 2.8
	SVM	50.6 ± 3.0	42.3 ± 1.7	56.2 ± 2.5

4.2.2 Text Categorization Data Set

Table 5 shows the performance of both the baseline approach and the CDM approach in different settings on the text categorization dataset. Both the kNN and RBF SVM classifiers with and without augmentation outperform the baseline approach.

Table 5: Means and standard deviations of classification accuracies (%) of the baseline and CDM approaches on the text categorization dataset by using 20 labeled training samples per class from the SM domain Spanish. The CDM approach shows the results in different settings.

LTM	Classifier	Baseline	Without augmentation	With augmentation
English	kNN	28.3 ± 11.4	66.5 ± 3.9	66.5 ± 3.9
	SVM	44.2 ± 3.3	65.3 ± 5.0	61.6 ± 3.9
French	kNN	28.3 ± 11.4	67.4 ± 4.5	67.4 ± 4.5
	SVM	44.2 ± 3.3	66.9 ± 6.1	61.4 ± 3.7
German	kNN	28.3 ± 11.4	67.4 ± 3.2	67.4 ± 3.2
	SVM	44.2 ± 3.3	69.3 ± 3.9	60.6 ± 3.8
Italian	kNN	28.3 ± 11.4	66.6 ± 2.8	66.6 ± 2.8
	SVM	44.2 ± 3.3	67.2 ± 3.7	60.8 ± 3.3

4.3 Influence of the Number of Training Samples per Class from SM domain

Figure 2 shows the accuracy of baseline and the proposed CDM approach with respect to the number of training samples per class for each data set from SM. For both the CDM and baseline approaches, we use the RBF SVM as the classifier. As shown in Fig. 2a and Fig. 2b, the accuracies of the baseline approach and CDM increase when we use a larger k .

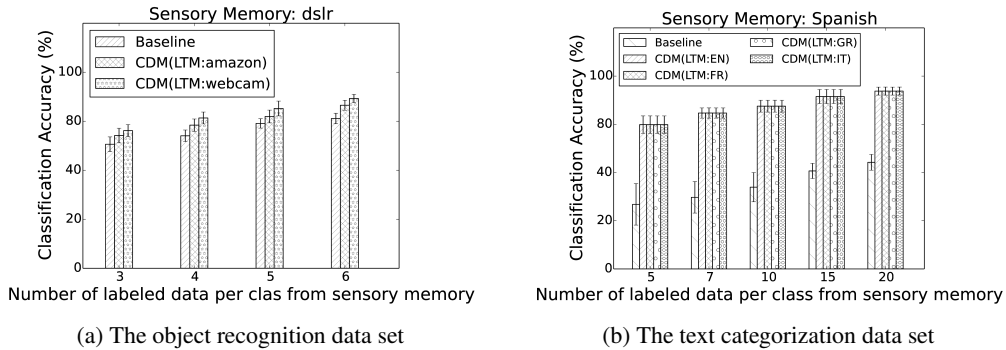


Figure 2: Classification accuracies of the baseline approach and CDM approach with respect to different numbers of SM training samples per class on the object recognition data set and the text categorization data set. In the case of CDM, different hatching patterns correspond to different LTM domains.

5 Conclusions

We proposed a computational model called CDM for the cognitive rapid learning problem. CDM has the following excellent properties: *a)* CDM is elegant, intuitive, and efficient; and, *b)* CDM has a complete theoretical architecture. The experimental results show that we can find at least one hypothesis or classifier that satisfies our hypothesis for rapid learning. One-shot learning or rapid learning is still a work in progress. From the CDM experience, we think that cognitive scientists should be required to pay attention to the role of information retrieved from LTM in rapid learning.

From the experimental results, we found that the feature augmentation did not always guarantee performance improvement. It is worth investigating this issue in the future.

In CDM, because we want to create a common metric space for LTM and SM, we proposed *global* metric learning in the working memory; unfortunately, this results in a CDM limitation. CDM attempts to find mapping matrices that minimize the sum of *all* pairwise distances between data points in the same class and maximize the sum of *all* pairwise distances between data points in different classes. As such, this implicitly assumes that classes form a single compact connected set. If the data instances are in highly multi-modal class distributions, the cost function will be penalized. Therefore, developing a discrimination approach that focuses on *local* neighborhoods for the problem is worth studying in future research.

References

- [1] Ralph Adolphs. The unsolved problems of neuroscience. *Trends in Cognitive Sciences*, 19(4):173–175, 2015.
- [2] Massih-Reza Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In *Proceedings of the 23rd Annual Conference on Advances in Neural Information Processing Systems (NIPS'09)*, pages 28–36, 2009.
- [3] Halneigh Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, pages 256–263, 2007.
- [4] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic Press Professional, Inc., 1990.
- [5] Steven C. H. Hoi, Wei Liu, and Shih-Fu Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, 2008.
- [6] Brenden Lake, Ruslan Salakhutdinov, and Joshua Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [7] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [8] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*, pages 213–226, 2010.
- [9] Robert J. Sternberg and Karin Sternberg. *Cognitive Psychology*. Wadsworth Publishing, sixth edition, 2011.
- [10] Anne Treisman. Features and objects in visual processing. *Scientific American*, 255(5):114–125, 1986.
- [11] Anne Treisman. Solutions to the binding problem: progress through controversy and convergence. *Neuron*, 24(1):105–125, 1999.
- [12] Janelle Weaver. How one-shot learning unfolds in the brain. *PLOS Biology*, 13(4):e1002138, 2015.

- [13] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 40–51, 2007.